

EXAM 2 – WRITTEN PORTION

NAME _____

SECTION NUMBER _____

CAMPUS MAILBOX NUMBER _____

EMAIL ADDRESS _____@rose-hulman.edu

Written Portion	/ 48
Computer Portion	/ 52
Total	/ 100

USE MATLAB SYNTAX FOR ALL PROGRAMS AND COMMANDS YOU WRITE

Problem 1: (4 points) Consider the code shown below. What is `vec` after running the code?

```
clc
clear variables
x = 1;
i = 1;
while x < 10
    vec(i) = 3*x;
    x = 2*x;
    i = i + 1;
end
```

Problem 2: You are given a matrix A :

$$A = \begin{bmatrix} 1 & 9 & 3 & 6 \\ 2 & 4 & 1 & 9 \\ 7 & 6 & 2 & 2 \\ 5 & 3 & 1 & 7 \end{bmatrix}$$

Write **one line** of code that will replace the fourth column of A with the numbers 1, 2, 3, 4.

Problem 3: (4 points) Below is a main code followed by a simple function:

```
% Main Code
i = 2;
j = 4;
k = 6;
[z] = simplefunction(i,j,k)

% Function Code
function [u] = simplefunction(r,s,t)
u = i +k/j;
```

After we run the main code, what is z in the main workspace?

- a) $z = 2$
- b) $z = 2.33333$
- c) $z = 3.5$
- d) $z = 5$
- e) An error occurs; explain what caused the error:

Problem 4: (4 points) What is the value of N after the following code is executed?

```
M = [1 2 3 4; 5 6 7 8; 9 10 11 12];
N = M(2:3,2);
```

Problem 5: (4 points) What is c after this code finishes running?

```
a = [1 4 16];  
b = [2 1 -1];  
c = sqrt(a).*(b.^2);
```

a. [4 2 -4]

f. [2 2 -4]

b. [4 4 16]

g. [4 4 -16]

c. [2 2 4]

h. The program crashes.

d. [4 2 4]

i. Other (explain): _____

e. [2 4 -16]

Problem 6: (4 points) The following code gives the error at right. Correct the error.

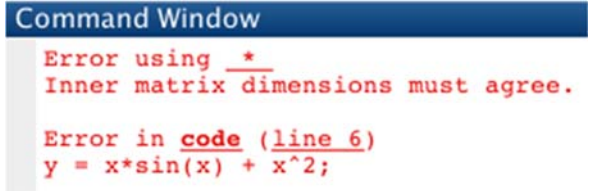
```
% Main Program  
dogs = 2;  
cats = 3;  
fish = 27;  
  
number_of_pets = pet_count(dogs,cats,fish);  
  
function [total] = pat_count(a,b,c);  
total = a + b + c;
```

```
>> Problem_6  
Undefined function or variable 'pet_count'.  
  
Error in Problem_6 (line 6)  
number_of_pets = pet_count(dogs,cats,fish);  
␣>> |
```

Problem 7: (4 points) Suppose we run the code snippet shown below from the MATLAB editor:

```
x = [0:0.1:3];  
y = x*sin(x) + x^2;
```

The code is supposed to create a vector of y values computed from the formula $y(x) = x \sin(x) + x^2$ for x values ranging from 0 to 3 in steps of 0.1. However, MATLAB instead returns the error shown. Fix the code so that, given the vector x , the vector y is generated in a single line of code (i.e., do **not** use a loop).



Command Window

```
Error using *  
Inner matrix dimensions must agree.  
  
Error in code (line 6)  
y = x*sin(x) + x^2;
```

Problem 8: (4 points) Consider the main code shown below that is followed by code for a function:

```
% Main code:  
a = 1;  
b = 2;  
c = 3;  
[x] = myFunction(a,b,c);  
[y] = myFunction(a,b,c);  
  
% Function code:  
function [X,Y] = myFunction(A,B,C)  
X = 3*A + C;  
Y = 4*B - 2*C;
```

We are supposed to get $x = 6$ and $y = 2$ when we run the main code, but we instead get $x = 6$ and $y = 6$. We know that the function `myFunction` itself behaves properly. Fix the code so it works as expected without modifying or duplicating `myFunction` in any way.

Problem 9: (4 points) The vectors height and width contain the height and width of 5 different rectangles. The code below is supposed to create a vector that contains the areas of these 5 rectangles, but it doesn't run correctly; it gives the error shown below instead. Fix the code so that it runs correctly.

```
height = [1 6 4 5 2];  
width = [2 8 5 6 3];  
area = height*width;
```

Command Window

```
Error using *  
Inner matrix dimensions must agree.
```

Problem 10: (4 points) We wrote a scrap of code to compute the sum of integers and the product of integers, for all those integers for which the product is less than 100:

```
clc  
clear variables  
i=1;  
sum=0;  
product=0;  
while product < 100  
    i=i+1;  
    sum=sum+i;  
    product=product*i;  
end
```

However, the program just runs forever. (It is in an infinite loop.) Mark the change on the code so that it runs correctly.

Problem 11: (8 points) We wish to write a function to compute the sum of the diagonal entries in a square matrix. (A square matrix has the same number of rows and columns. A diagonal entry is an element of the matrix that has the same row index and column index.) For example, for the matrix

```
>> a=[1 0 -2; 2 3 5; 7 2 11]
```

a =

```
    1     0    -2
    2     3     5
    7     2    11
```

The sum of the diagonal entries is $1+3+11=15$. The function header is given below—write the code to complete the function.

```
function [sumval]=tr(a)
```

EXAM 2 – COMPUTER PORTION

For this exam, you will create **TWO** m-files: a main script called `lastname_firstname_exam_2.m` (all lower case) and a function called `lastname_circle.m` (all lower case).

The header section of your code should include your name, section number, and CM number; you do not need to do any other commenting. **There should be no output other than what is requested.**

Problem 1 (17 pts)

In your main script, add code to create a table of the powers of 2 (i.e. $y=2^n$), so long as the power of 2 is less than 1000. You must use a while loop to accomplish this task. The first part of the table should look like:

n	y
1	2
2	4
3	8
4	16
5	32

The last entry in the table for y should be less than 1000.

(over)

Problem 2 (35 pts)

(a) (10 points) Create a function with the following function header:

```
function lastname_circle(centerx,centery,radius,angle1,angle2);
```

The purpose of the function is to plot a circular arc or circle. Recall that points on a circular arc or circle are defined by the equations

$$x = x_c + r * \cos(\theta)$$
$$y = y_c + r * \sin(\theta)$$

Here are the meanings of the variables and inputs:

- x_c (`centerx`) is the x center of the arc or circle
- y_c (`centery`) is the y center of the arc or circle
- r (`radius`) is the radius of the arc or circle
- θ is the angle in degrees in the equations. For the function, `angle1` and `angle2` are the starting and ending angles in degrees for the arc or circle. For example, if `angle1` is 0 and `angle2` is 360 you will have a circle but if `angle1` is 0 and `angle2` is 180 you will have a half-circle (circular arc).

Complete the function so that it draws appropriate circular arcs or circles.

- (b) (5 points) Add code to your main script to call the function and draw a circle centered at $x=0$ and $y=2$, with a radius of 1. Add the `axis equal` command to your main script to make the circle appear round.
- (c) (10 points) Download the spreadsheet 'Bear_Data.xlsx' from Moodle. It contains data for a picture made entirely of circles. We will create the picture by repeatedly calling the `lastname_circle` function. The first column of the data is the x center, the second column of the data is the y center, the third column contains the radius, and the fourth and fifth columns are `angle1` and `angle2`. Add code to your main script to make a new figure, read in the data from the spreadsheet, and create the new picture. Do not hardcode the data for the picture! Give the picture the title "Happy Bear".
- (d) (10 points) Now we would like to modify the function and the main routine to make circles filled with colors and arcs filled with colors. (Hint: review Day 23 materials.) There is a 6th column in the spreadsheet that contains a color code for each circle: 1=red, 2=green, 3=blue. Modify your function to accept another input that is the color code, and modify your main routine to send the appropriate color code to the function. Within the function add the code necessary to create color-filled arcs and circles. Revise the picture title to say "Happy Blue Bear".

When you are finished, put your main program (`lastname_firstname_exam_2.m`) and your function (`lastname_circle.m`) in the Moodle assignment dropdown.

NOTE: All programming must stop 5 minutes before the end of the period. You will have 5 minutes after that to upload your files to Moodle if you need that time.